Design and Implementation of Security Gateway for Synchrophasor based Real-time Control and Monitoring in Smart Grid

Khan, R., McLaughlin, K., Laverty, D., & Sezer, S. (2017). Design and Implementation of Security Gateway for Synchrophasor based Real-time Control and Monitoring in Smart Grid. *IEEE Access*, *5*, 11626-11644. https://doi.org/10.1109/ACCESS.2017.2716440

Published in:
IEEE Access

Document Version:
Publisher's PDF, also known as Version of record

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 14, 2017, accepted June 6, 2017, date of publication June 28, 2017, date of current version July 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2716440

Design and Implementation of Security Gateway for Synchrophasor Based Real-Time Control and Monitoring in Smart Grid

RAFIULLAH KHAN, KIERAN MCLAUGHLIN, DAVID LAVERTY, AND SAKIR SEZER
Queen's University Belfast, Belfast BT7 1NN, U.K.
Corresponding author: Rafiullah Khan (rafiullah.khan@qub.ac.uk)

This work was supported by the EPSRC CAPRICA Project under Grant EP/M002837/1.

ABSTRACT Synchrophasor technology has numerous applications ranging from simple grid monitoring/visualization to real-time protection and control. Most legacy phasor measurement units (PMUs) and phasor data concentrators (PDCs) deployed in power grids support the IEEE C37.118.2 communication framework, which is highly vulnerable to cyber attacks due to lack of inherent security mechanisms. The IEC 61850-90-5 recently emerged as new communication framework with support for security features but its use in commercial devices is still very limited. The replacement of legacy PMUs/PDCs in power grids is a big challenge due to cost and deployment complexity. The concept of a gateway has recently been proposed in the literature to enable IEEE C37.118.2 compatible PMUs to send data in IEC 61850-90-5 format. However, the published gateway has limited features and also lacks security functionalities. This paper addresses security, interoperability, and integration issues between legacy and state-of-the-art phasor devices through the design of a security gateway. The security gateway is implemented with flexibility in mind and can be used for PMUs as well as PDCs under different configurations. It provides: 1) protocol conversion functionalities (from IEEE C37.118.2 to IEC 61850-90-5 and vice versa) and 2) security functionalities based on IEC recommended group domain of interpretation security mechanism. The security gateway is very compact in size, based on low-power ARM processor and inexpensive to be deployed in power systems. Through detailed experimental evaluation with real PMU data, this paper also validated the suitability of the security gateway for different types of synchrophasor applications with strict latency and data rate requirements.

INDEX TERMS Smart grid, synchrophasors, cyber security, IEEE C37.118.2, IEC 61850-90-5, GDOI.

I. INTRODUCTION

Synchrophasor technology involves measurement of electrical quantities using Phasor Measurement Units (PMUs) at different locations in power grids which ideally will be transmitted in real-time to the control center. The PMUs are equipped with a GPS antenna to time-stamp measured electrical quantities and enable the control center to precisely process them for real-time Wide-Area Monitoring, Protection And Control (WAMPAC) of power grids. Synchrophasor technology requires an efficient and secure communication framework to transfer power system dynamics in real-time over IP network to ensure reliable and trustworthy WAMPAC operation.

IEEE C37.118.2 and IEC 61850-90-5 are two well known synchrophasor communication frameworks [1]. IEEE C37.118.2 emerged from IEEE C37.118 (developed in 2005) and has been widely adopted in commercial PMUs and Phasor Data Concentrators (PDCs). However, it has several gaps e.g., no inherent security mechanism. To harmonize synchrophasor data transfer with IEC 61850 power utility automation standard and fulfill gaps of IEEE C37.118.2, IEC 61850-90-5 was introduced in 2012 with a recommended security mechanism based on Group Domain of Interpretation (GDOI) [2]. Until now, few commercial phasor devices support IEC 61850-90-5.

A. PAPER MOTIVATION

This paper addresses two key challenges faced today in synchrophasor-based systems: (i) interoperability and integration between legacy and state-of-the-art phasor devices, and (ii) security of synchrophasor data communication.

Over the last two decades, many power companies have already deployed hundreds or thousands of PMUs and PDCs which lack the recent IEC 61850-90-5 communication

framework [3] and/or the GDOI security mechanism [2]. These legacy devices use different protocol and data representations, which make interoperability and integration a major issue. The interoperability constraints make the adaptation of secure IEC 61850-90-5 in power grids a big challenge. Upgrading or replacement of legacy PMUs, PDCs and previously developed IEEE C37.118.2 compatible WAMPAC applications is not feasible due to the cost and complexity involved.

Depending on the type of synchrophasor application, particularly new real-time control applications, a cyber attack could result in catastrophic consequences for the power grids. WAMPAC applications based on IEEE C37.118.2 are highly vulnerable and different types of cyber attacks have been investigated/demonstrated in literature [4]–[6]. Even though, GDOI is recommended as a security mechanism for synchrophasor data communication, most commercial phasor devices (including many IEC 61850-90-5 compliant PMUs and PDCs) lack its implementation. Without proper authentication, encryption and cryptographic signature, IEC 61850-90-5 is as vulnerable to cyber attacks as IEEE C37.118.2. Unless this security issue is addressed, synchrophasors can never be trusted to support emerging real-time control techniques.

B. PAPER CONTRIBUTIONS

A low cost, easily deployable and integrable solution is required to address challenges presented in Section I-A. Recently, Firouzi *et al.* [7] identified interoperability issues between legacy and state-of-the-art PMUs and proposed a solution based on an IEEE C37.118.2 to IEC 61850-90-5 gateway. However, their published work has several limitations including: (i) the gateway lacks GDOI security mechanism to ensure communication security, (ii) the gateway supports one way conversion from IEEE C37.118.2 to IEC 61850-90-5 but not vice versa which is also essential to properly address interoperability issues (e.g., an IEEE C37.118.2 compliant WAMPAC application), (iii) the gateway functionalities were demonstrated for a PMU but lacks support for multiple phasor devices (especially PDCs under different configurations), (iv) detailed experimental evaluation is missing to validate the suitability of gateway for PMUs and PDCs in different synchrophasor applications with strict latency and data rate requirements.

To properly address the key challenges (presented in Section I-A) and overcome the limitations of the best previous work [7], this paper describes the implementation of a flexible security gateway that is compliant with both, IEEE C37.118.2 and IEC 61850-90-5 communication frameworks. The security gateway is capable to receive synchrophasor data from PMUs/PDCs in IEEE C37.118.2 or insecure IEC 61850-90-5 format, convert it into secure IEC 61850-90-5 using GDOI security mechanism and transfer over wide-area network to the control center. Further, a security gateway in control center converts back secure IEC 61850-90-5 packets into the original insecure IEEE C37.118.2 or insecure IEC 61850-90-5 packets. The conversion feature from insecure IEC 61850-90-5 into secure IEC 61850-90-5 and vice versa is essential as many IEC 61850-90-5 compliant commercial phasor devices, applications and even tools (e.g., PMU Connection Tester) do not support security features. The security policies and keying material used by the security gateway are not fixed and replaced periodically. This provides strong protection against cryptanalysis as any cracked security credentials will longer remain valid. In short, key contributions of this paper include:

1) Design and implementation of GDOI security mechanism. This paper provides clear functional specification towards practically implementing GDOI security mechanism as IEC 61850-90-5 [3] and RFC 6407 [2] lacks sufficient technical implementation guidelines.
2) Due to no open-source availability of communication frameworks (particularly IEC 61850-90-5), IEEE C37.118.2 libraries and IEC 61850-90-5 libraries were implemented and validated for the correctness of functionalities with the PMU Connection Tester.
3) Implementation of the security gateway using implemented IEEE C37.118.2 libraries and IEC 61850-90-5 libraries and the integration of GDOI security mechanism.
4) Functional analysis of the security gateway. Note that Wireshark lacks support for IEC 691850-90-5 communication framework. Thus, a Wireshark dissector was also implemented and utilized in the functional analysis.
5) Detailed performance evaluation of GDOI security mechanism and the security gateway has been performed in real networking environment. It is demonstrated that an effective low cost and compact implementation of the security gateway is feasible using an ARM processor.
6) Experimental evaluation of the security gateway with real PMU, emulated PMU (ePMU) and different types of emulated PDCs (ePDCs) under different configurations. It is experimentally validated that the security gateway is suitable for different types of PMUs and PDCs in different categories of synchrophasor applications which have strict latency and data rate requirements.

C. PAPER ORGANIZATION

The paper is organized as follows: Section II presents background and related work from literature. Section III presents the design of the security gateway including GDOI security mechanism. Section IV addresses implementation of the security gateway. Section V practically evaluates the security gateway in real networking environment and presents performance metrics. Finally, Section VI concludes the paper.

II. BACKGROUND & RELATED WORK

Synchrophasor technology has great potential to play a vital role in WAMPAC applications [8]. It can enable operators

or control algorithms to take prompt actions through real-time tracking of power system dynamics/disturbances. It consists of two important components: PMUs and PDCs. PMUs measure electrical quantities and transmit them to the control center. PDCs aggregate data from multiple PMUs (and/or other PDCs) based on GPS timestamp and send as single output stream. The number of PMUs aggregated by a PDC depends on the type and location (e.g., substation PDC: 20-40 PMUs, regional center PDC: 50-500 PMUs, super PDC: 500+ PMUs) [9].

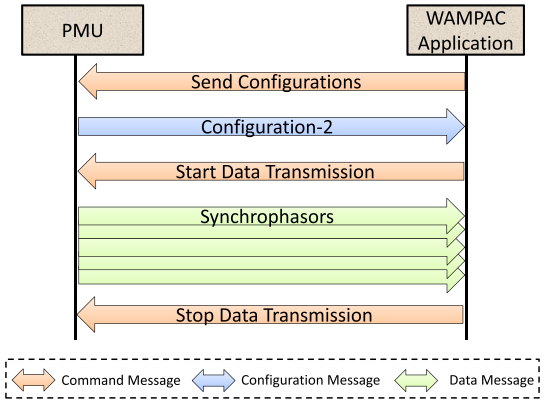


FIGURE 1. IEEE C37.118.2 communication scenario.

Most legacy phasor devices support IEEE C37.118.2 communication framework which consists of four types of messages: data (sent by PMU and contains actual synchrophasor data), configuration (sent by PMU and contains PMU configurations), command (received by PMU and contains instructions/orders) and header (sent by PMU and contains descriptive information in human readable format). A basic communication scenario based on IEEE C37.118.2 is depicted in Fig. 1. IEEE C37.118.2 has no restriction on communication mode and transport protocol but lacks security mechanism [10]. IEC 61850-90-5 originally evolved from IEC 61850 by modifying the two Ethernet protocols: (i) Generic Object Oriented Substation Events (GOOSE) and (ii) Sampled Value (SV)). As depicted in Fig. 2, the new GOOSE and SV protocols are routable and operate over transport layer and network layer protocols. The modified protocols are also known as Routable-GOOSE (R-GOOSE) and Routable-SV (R-SV). R-SV being a stream based protocol is the most suitable choice for synchrophasor communication. As shown in Fig. 2, IEC 61850-90-5 also include a GDOI security mechanism for protecting R-GOOSE and R-SV communication. It is notable that several recent commercial PMUs support R-SV but lack GDOI security mechanism. Authors in [11] explained challenges in commissioning of IEC 61850-90-5 and its comparison with IEEE C37.118.2 is presented in [1]. An OpenPMU project is presented in [12] that is compliant with both IEEE C37.118.2 and IEC 61850-90-5.

IEEE C37.118.2 and IEC 61850-90-5 without GDOI security mechanism are highly vulnerable to cyber

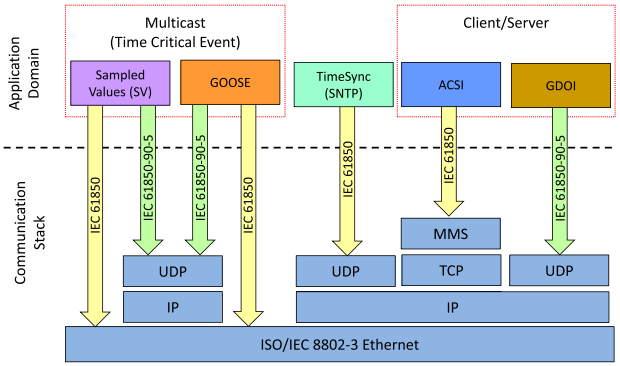


FIGURE 2. IEC 61850-90-5 protocol stack.

attacks. Several researchers have investigated vulnerabilities especially in IEEE C37.118.2 [4]–[6], [10]. Authors in [6] and [13] investigated different types of cyber attacks on synchrophasor based systems. Authors in [14] performed Denial of Service (DoS) attack and measured PMU resilience when flooded with ARP requests and IPv4 packets. They also measured resilience against malformed packets through protocol mutation experiments. Authors in [15] suggested firewall and Virtual Private Network (VPN) to protect synchrophasor communication from cyber attacks. Authors in [16] analyzed impact of packet drop attacks on synchrophasor based systems and proposed a mechanism for detecting such attacks. Authors in [17] analyzed data integrity attacks in synchrophasor-based WAMPAC applications. In short, synchrophasor-based systems without a security mechanism are highly vulnerable to cyber attacks. Security challenges for synchrophasors and smart grid in general have been addressed in several survey articles [18], [19].

Firouzi *et al.* [7] proposed a low cost gateway that converts IEEE C37.118.2 packets into IEC 61850-90-5. However, their work lacks several features necessary for properly addressing interoperability issues between legacy and state-of-the-art phasor devices and WAMPAC applications (e.g., an IEEE C37.118.2 compliant WAMPAC application). Further, their gateway lacks security features vital for protection of synchrophasor-based systems from cyber attacks. Thus, this paper presents a security gateway which is very flexible and works under different configurations of PMUs as well as PDCs. The security gateway not only provides interoperability between legacy and state-of-the-art phasor devices but also ensures protection against cyber attacks by using the GDOI security mechanism.

III. DESIGN OF SECURITY GATEWAY

The security gateway acts as a secure proxy between two otherwise inherently insecure end devices. The basic scenario is depicted in Fig. 3. The security gateway can be a dedicated device to each PMU/PDC or it can offer security services to a group of PMUs and PDCs. In the substation, it receives packets from PMU/PDC in insecure IEEE C37.118.2 or

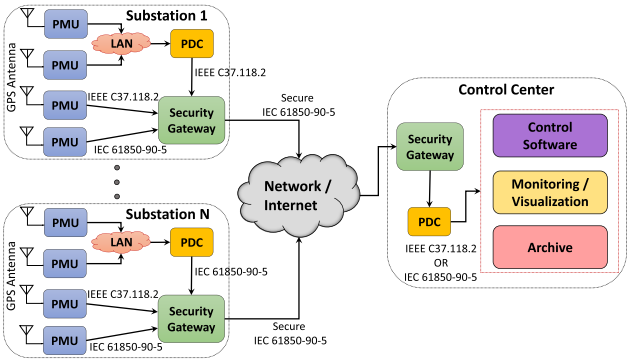


FIGURE 3. Security gateways in synchrophasor-based system.

insecure IEC 61850-90-5 format, encrypts them, applies cryptographic signatures and transmits them to the security gateway in control center as secure IEC 61850-90-5. The security gateway in control center decrypts the packets, verifies signatures and provides the packets in insecure IEEE C37.118.2 or insure IEC 61850-90-5 format to monitoring, control or archiving application. Thus, the security gateway is flexible, configurable and equally suitable for any legacy PMU/PDC or WAMPAC application which support either IEEE C37.118.2 or IEC 61850-90-5 communication framework.

The security gateway is intelligent enough that if a secure IEC 61850-90-5 packet (i.e., has encryption or cryptographic signature or both) is received from PMU/PDC in substation, it simply forwards the packet to control center without applying encryption and cryptographic signature. This feature is useful if a recent PMU/PDC is deployed in substation that supports security features. Based on the security information inside IEC 61850-90-5 packets, security gateway in control center identifies such packets and forwards to WAMPAC application without processing. The security gateway is also intelligent enough to differentiate between received secure and insecure packets. If the security gateway in the control center receives insecure packets (i.e., have no encryption and cryptographic signature) in IEEE C37.118.2 or IEC 61850-90-5 format, it forwards them to control/monitoring application without processing. This feature is useful if the security gateway is deployed for some legacy PMUs/PDCs but not for all. In short, the security gateway in the control center only processes a packet if it has both, encryption as well as cryptographic signature applied by a security gateway of a PMU/PDC. The security policies and keying material used by the security gateways are not fixed but instead replaced periodically based on the GDOI security mechanism (i.e., a recommended security mechanism for synchrophasor-based systems by IEC [3]).

This section provides an insight of the security gateway functionalities in order to highlight main architectural components need to be implemented. In short, the design of security gateway involves two challenges: (i) mapping of IEEE C37.118.2 packets into IEC 61850-90-5 and vice versa (addressed in Section III-A), and (ii) authentication and periodic update of security polices and keying material based on GDOI security mechanism (addressed in Section III-B).

A. MAPPING BETWEEN IEEE C37.118.2 AND IEC 61850-90-5

The security gateway can work in four possible scenarios: (i) PMU/PDC is IEEE C37.118.2 compliant, (ii) PMU/PDC is IEC 61850-90-5 compliant, (iii) WAMPAC application is IEEE C37.118.2 compliant and (iv) WAMPAC application is IEC 61850-90-5 compliant.

1) IEEE C37.118.2 COMPLIANT PMU/PDC

In this case, the security gateway receives packets from a PMU/PDC in IEEE C37.118.2 format, maps them into secure IEC 61850-90-5 format and transmits them to the control center. To enable the security gateway to successfully map IEEE C37.118.2 data messages into IEC 61850-90-5 R-SV messages, certain packets are exchanged between a PMU/PDC and a security gateway. As depicted in Fig. 1, the receiver needs configuration message type 2 (CFG-2) which contains necessary information on how to decode upcoming data messages. In short, the security gateway establishes communication with a PMU/PDC as follows: *Step-1:* security gateway sends command message to PMU/PDC and requests CFG-2, *Step-2:* PMU/PDC provides CFG-2 to security gateway, *Step-3:* security gateway sends command message to PMU/PDC and requests to start synchrophasor data transmission, *Step-4:* PMU/PDC starts continuous streaming of data messages (which contain actual synchrophasor data) to security gateway, and *Step-5:* security gateway sends command message to PMU/PDC whenever necessary and requests to stop synchrophasor data transmission. Note that the security gateway converts only a continuous stream of data messages into secure IEC 61850-90-5 R-SV format using CFG-2 configurations.

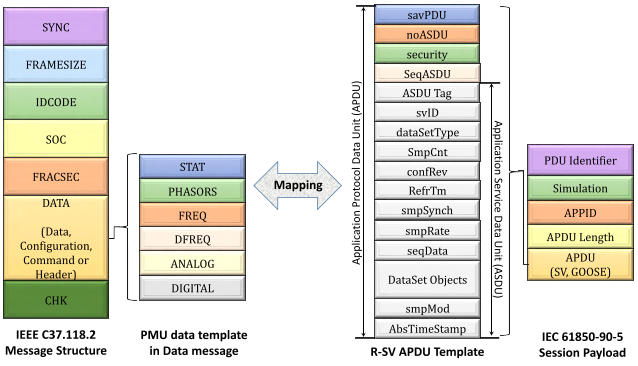


FIGURE 4. Synchrophasor data mapping in IEEE C37.118.2 data message and IEC 61850-90-5 R-SV message.

Fig. 4 illustrates mapping between IEEE C37.118.2 data message and IEC 61850-90-5 R-SV message. For sake of simplicity, Fig. 4 does not depict IEEE C37.118.2 CFG-2 configuration message, IEC 61850-90-5 R-SV session

header/information and underlying layers (i.e., transport, network, link layers). FREQ, DFREQ and size, format and number of PHASORS, ANALOG AND DIGITAL values in PMU data template in IEEE C37.118.2 data message (in Fig. 4) are determined by the security gateway using CFG-2 configurations. For PDC, IEEE C37.118.2 data message contains multiple PMU data templates (one for each PMU inside a PDC). The same settings can be reflected in IEC 61850-90-5 as a R-SV session payload (in Fig. 4) can support multiple Application Protocol Data Units (APDUs) and Application Service Data Units (ASDUs) inside same packet. PHASORS, ANALOG AND DIGITAL values in IEEE C37.118.2 data message can be mapped into IEC 61850-90-5 R-SV dataset objects using appropriate data types [3]. Before mapping, it is necessary to apply correct conversion factors (i.e., PHUNIT, ANUNIT, DIGUNIT) specified in CFG-2. Note that FREQ in IEEE C37.118.2 data message represents frequency deviation from nominal in mHz. To calculate actual frequency, nominal line frequency (i.e., FNOM) information should be taken into account from CFG-2. To determine the timestamp from IEEE C37.118.2 data message, fractional time and SOC (i.e., second of century) should be added. Fractional time is calculated by dividing first 24 bits of FRACSEC by the TIME_BASE (specified in CFG-2). Note that R-SV APDU template in Fig. 4 is ASN.1 encoded i.e., every field has Tag-Length-Value (TLV) format. ASN.1 encoding helps uniquely identify each field and its size. Detailed description of PMU/PDC modeling into IEC 61850-90-5 logical devices and nodes is out of scope for this paper and has been provided in [3], [7], and [20]. The security gateway encrypts the IEC 61850-90-5 session payload and applies cryptographic signature on entire session layer data (excluding signature itself) before transmitting the R-SV packet to the control center. The session header also contains security information such as IDs of encryption and signature algorithms, key ID, key validity etc. The security features are based on GDOI security mechanism and are addressed in Section III-B.

2) IEC 61850-90-5 COMPLIANT PMU/PDC

In this case, the security gateway may receive secure or insecure IEC 61850-90-5 packets (depending on whether a PMU/PDC supports a security mechanism). The security gateway forwards packets to the control center and applies encryption and cryptographic signature to only insecure IEC 61850-90-5 packets. This scenario is computationally less complex for the security gateway compared to IEEE C37.118.2 compliant PMU/PDC as received packets are not fully decoded. The security gateway decodes session header and reads KeyID, TimeofCurrentKey, TimetoNextKey and SecurityAlgorithms (i.e., IDs of encryption and signature algorithms). If SecurityAlgorithms indicates that no encryption and no signature is used, the security gateway encrypts the session payload and calculates/adds signature to the packet. It also specifies KeyID, TimeofCurrentKey, TimetoNextKey and SecurityAlgorithms in session header based on the security credentials applied to that packet.

3) IEEE C37.118.2 COMPLIANT WAMPAC APPLICATION

In this case, the security gateway in the control center receives secure IEC 61850-90-5 R-SV packets from the security gateway of the substation (as depicted in Fig. 3), verifies the signature, decrypts the packets, converts them into IEEE C37.118.2 data messages and transmits them to the control/monitoring/archiving application. The start of communication between security gateway and control center WAMPAC application follows same IEEE C37.118.2 semantics as depicted in Fig. 1. In current implementations, the security gateway can be configured to provide CFG-2 configurations and data messages to the WAMPAC application without being requested through IEEE C37.118.2 command messages.

The mapping of IEC 61850-90-5 R-SV packets into IEEE C37.118.2 data messages is illustrated in Fig. 4 and follows the reverse process of that described in Section III-A.1. Note that CFG-2 configurations are now created by the security gateway and provided to the control center WAMPAC application. As CFG-2 configurations contain information on how to decode upcoming data messages, the security gateway must generate and send new CFG-2 message to the WAMPAC application each time it changes scaling/conversion factors or structure of the IEEE C37.118.2 data messages.

4) IEC 61850-90-5 COMPLIANT WAMPAC APPLICATION

In this case, the security gateway in the control center receives secure IEC 61850-90-5 R-SV packets from the security gateway of the substation (as depicted in Fig. 3), verifies the signature, decrypts the packets and transmits insecure IEC 61850-90-5 R-SV packets to the control/monitoring/archiving application. In this case, the security gateway does not need to decode entire session payload/APDUs to reduce computational complexity. To verify the signature and decrypt a packet, the security gateway utilizes security information specified in the same packet session header. It is necessary that the security gateway resets security information (i.e., KeyID, TimeofCurrentKey, TimetoNextKey and SecurityAlgorithms) in newly generated insecure IEC 61850-90-5 R-SV packets before transmitting them to the control center WAMPAC application.

If security information inside a received secure IEC 61850-90-5 R-SV packet does not match the valid security credentials provided by GDOI security mechanism, security gateway simply forwards same packet to WAMPAC application without any changes/processing. This indicates that the PMU applied security features on this packet but not the security gateway of the substation. However, such scenario is rare in practice as most commercial PMUs deployed in power systems lack security features as well as IEC 61850-90-5 communication framework.

B. GDOI BASED SECURITY MECHANISM

GDOI is a group security framework and published by Cisco Systems and MIT in RFC 6407 [2]. It is recommended

security mechanism for synchrophasor-based systems by IEC [3] due to its three key features: (i) authentication, (ii) freshness and (iii) secrecy. Authentication ensures that only allowed group member devices are able to acquire security credentials and securely communicate among each other. Freshness provides protection against cryptanalysis attacks by periodically replacing/updating security credentials. It also ensures that allowed devices could not be tricked into accepting old/expired security credentials. GDOI ensures two types of freshness: recency freshness (i.e., an authenticated device should accept only the most recent security credentials which are valid for certain duration) and sequential freshness (i.e., an authenticated device should never accept security credentials it already has or used in the past). GDOI ensures perfect forward secrecy i.e., if a security key is compromised, the adversary can only acquire new security keys distributed under the protection of compromised key. However, past communication still remains protected and cannot be decrypted by an adversary. The GDOI security mechanism also provides forward and backward access control. Thus, a device has no access to valid/current security credentials after leaving the group (i.e., forward access control) and before joining the group (i.e., backward access control). Every time a device leaves the group, GDOI security mechanism generates new security credentials and distributes to the remaining group members.

The GDOI security mechanism consists of two phases: *phase-1:* authentication and *phase-2:* periodic security credentials update. RFC 6407 [2] addresses only phase-2 but lacks details about phase-1 which is based on Internet Security Association and Key Management Protocol (ISAKMP). Further, IEC in [3] has briefly addressed minor changes in GDOI phase-2 but lacks clear functional specification towards practically implementing it. Thus, this section provides clear functional understanding of GDOI security mechanism for implementing it in the security gateway.

1) FUNCTIONAL OVERVIEW OF SECURITY MECHANISM

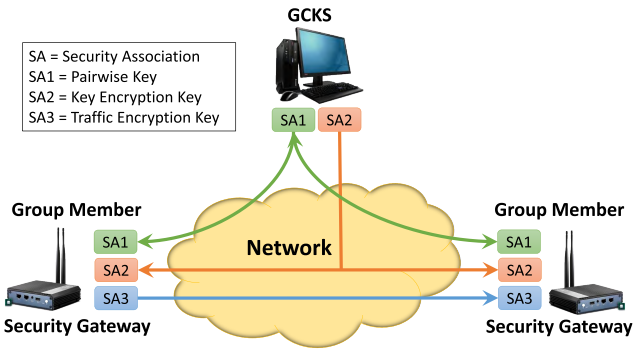


FIGURE 5. **Functional overview of GDOI security mechanism. Note that the security gateway works as a group member of GCKS.**

Fig. 5 depicts a functional overview of the GDOI security mechanism. It consists of two types of devices: Group Controller/Key Server (GCKS) and Group Member (GM). As GDOI is a group security mechanism, GCKS manages security policies and keying material for a certain number of GMs. Based on the proposed system in Fig. 3, each security gateway acts as a GM of GCKS. After acquiring security credentials from GCKS, GMs can securely communication among each other. To ensure communication among GMs remains highly secure, GCKS periodically updates group security policies and keying material. It is possible in GDOI that a single GCKS manages multiple groups (each group with different security policies and keying material) or a single group is managed by multiple GCKSs. This feature is useful if a security gateway is used in more than one synchrophasor application. In current implementations, GCKS can manage only one group and assumes that each GM device is part of only one group.

After GDOI phase-1 (i.e., authentication between a GM and GCKS), GDOI phase-2 consists of two types of exchanges: (i) GroupKey-Pull and (ii) GroupKey-Push. During GroupKey-Pull, a GM requests group security policies and keying material from GCKS. Whereas in GroupKey-Push, GCKS distributes updates of group security policies and keying material to all authorized GMs. In short, the functionalities of GCKS include: (i) generate and manage group security policies and keying material, (ii) process authentication requests from GMs, (iii) perform GroupKey-Pull exchange with GMs, (iv) admit a GM or expel it from the group, and (v) declare new security credentials as current and cause old security credentials to expire (even before its validity if necessary). The actions of a GM are limited and include: (i) initiate authentication with GCKS, (ii) initiate GroupKey-Pull exchange with GCKS, and (iii) accept updates of security credentials through GroupKey-Push.

2) TYPES OF SECURITY POLICIES AND KEYING MATERIAL

As depicted in Fig. 5, GDOI security mechanism consists of three types of security policies and keying material: (i) Pairwise key, (ii) Key Encryption Key (KEK) and (iii) Traffic Encryption Key (TEK). Pairwise security credentials are specific to each GM and are used during its authentication with GCKS to join a specific group (i.e., GDOI phase-1). Pairwise key also protects GroupKey-Pull exchange used to acquire KEK and TEK from GCKS. KEK and TEK security policies and keying material are generated by GCKS and are specific to GDOI phase-2. KEK protects GroupKey-Push exchange in which GCKS provides updates of security policies and keying material (i.e., new KEK and new TEK). Whereas, TEK security credentials are used to protect communication among GMs.

3) AUTHENTICATION AND PAIRWISE SECURITY CREDENTIALS

RFC 6407 [2] does not specify mutual authentication and authorization between GM and GCKS. The choice of authentication technique, pairwise security policies and keying material depends on the developers. The authentication process can be based on certificates, pre-shared keys or any other

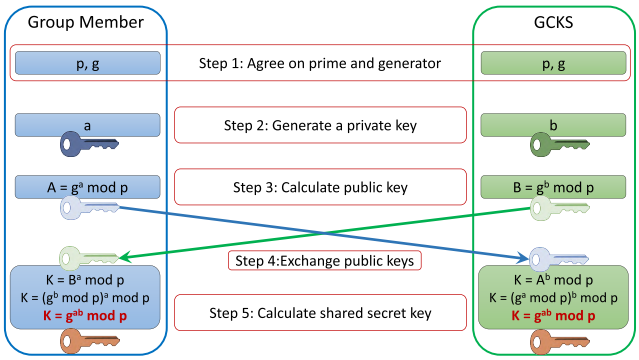


FIGURE 6. Diffie Hellman mechanism for pairwise key establishment between GCKS and a group member.

public key cryptography technique. This paper addresses GDOI phase-1 authentication based on authorization list (i.e., verifying GM's identity) and Diffie Hellman public key cryptography technique. Thus, the authentication phase is very light-weight, flexible, scalable and specially suitable for constrained devices.

Fig. 6 illustrates the establishment of pairwise key between a GM and GCKS using Diffie-Hellman public key cryptography technique. It can be observed that both devices have no prior knowledge about the secret key and establish it jointly. It can be observed in Fig. 6 that GM and GCKS first agree on a prime, p and generator, g. They generate a private key and calculate public key from it using prime and generator in a specific mathematical model. The public keys can take any value between 1 to p-1. Both devices exchange their public key with one another and derive a common shared secret key from it using a specific mathematical model. This key is used as pairwise key to protect GDOI phase-2 in current implementations. The shared key has two features: (i) it cannot be generated by a device alone but can only be generated jointly by interacting with other device, and (ii) no third party (that is not involved in the communication) can deduce the shared secret key. It is worth to mention that messages exchanged in Diffie-Hellman are unencrypted. However, any eavesdropping on the communication cannot enable adversaries to deduce the secret key. The larger the values of private keys and prime, the more secure will be Diffie-Hellman exchange. Different Oakley groups are defined for Diffie-Hellman exchange which are mainly different in how the prime is generated [21]. Current implementations are based on default Oakley group with 768 bits prime number.

Three different types of ISAKMP authentication techniques are available for exchange of public keys (i.e., step 4 in Fig. 6): (i) Base Exchange (BE), (ii) Identity Protection Exchange (IPE) and (iii) Aggressive Exchange (AE). They are mainly different in terms of number of packets exchanged, their structure and encryption as depicted in Fig. 7. The format/structure of each payload type in Fig. 7 is defined in [21]–[23] and is out of scope of this paper. AE is the fastest authentication technique due to exchange of very few packets. The key limitation of BE and AE is the lack of identity protection of communicating devices from eavesdroppers. Thus, the implementations in this paper targeted IPE even though it is comparatively a bit slower authentication technique. The IPE (as depicted in Fig. 7(b)) consists of six steps: *step-1:* GM offers security proposal or multiple proposals (e.g., encryption algorithm, authentication algorithm, key type, key validity etc) to GCKS, *step-2:* GCKS chooses appropriate proposal and responds to the GM with accept/reject decision, *step-3:* GM sends its Diffie Hellman public key and related data to GCKS, *step-4:* GCKS provides its own Diffie Hellman public key to GM, *step-5:* GM calculates shared secret key (i.e., step 5 in Fig. 6), uses it to encrypt its identification information and sends to GCKS, and *step-6:* GCKS also calculates shared secret key, uses it to verify GM's identity and provides its own identification information to GM.

4) GroupKey-PULL EXCHANGES

GroupKey-Pull is initiated by GMs to acquire security policies and keying material (i.e., KEK and TEK) from GCKS under the protection of GDOI phase-1 pairwise security credentials. The goal of GroupKey-Pull is to establish rekeying with GCKS and acquire security credentials for secure communication among group members (i.e., security gateways). GroupKey-Pull uses nonces to guarantee liveness and achieve protection against replay attacks. GDOI GroupKey-Pull consists of total four steps/messages as depicted in Fig. 8: *step-1:* a GM requests KEK and TEK security policies from GCKS, *step-2:* GCKS responds to GM with the supported security policies, *step-3:* GM sends key download request to GCKS, and *step-4:* GCKS responds to GM with KEK and TEK and their security policies. The format/structure of each payload type in Fig. 8 is defined in [2]. It is worth to mention that some payload types have different structure than ISAKMP payloads in Fig. 7. Further, the structure of KD payload (which contains actual KEK and TEK security policies and keying material) is slightly different from [2] and presented by IEC in [3].

It can be observed in Fig. 8 that each message has a unique HASH calculation method according to RFC 2409 [21]. Each HASH is calculated using pseudo-random function (prf) over certain data as follows:

$$\text{HASH}(1) = \text{prf}(\text{SKEYID\_a}, \text{M-ID} \mid \text{Ni} \mid \text{ID})$$
$$\text{HASH}(2) = \text{prf}(\text{SKEYID\_a}, \text{M-ID} \mid \text{Ni\_b} \mid \text{Nr} \mid \text{SA})$$
$$\text{HASH}(3) = \text{prf}(\text{SKEYID\_a}, \text{M-ID} \mid \text{Ni\_b} \mid \text{Nr\_b} \mid \text{GAP})$$
$$\text{HASH}(4) = \text{prf}(\text{SKEYID\_a}, \text{M-ID} \mid \text{Ni\_b} \mid \text{Nr\_b} \mid \text{SEQ} \mid \text{KD})$$

where SKEYID_a is GDOI phase-1 secret and M-ID is message ID. Ni and Nr are initiator (i.e., GM) and responder (i.e., GCKS) nonce payloads, respectively. Ni_b and Nr_b are nonce values passed inside Ni and Nr, respectively. The SEQ ensures anti-replay state and protects GM from accepting GroupKey-Push message sent prior to joining the group.

5) GroupKey-PUSH EXCHANGE

GCKS provides security policies and keying material updates to GMs in GroupKey-Push exchange. GroupKey-Push

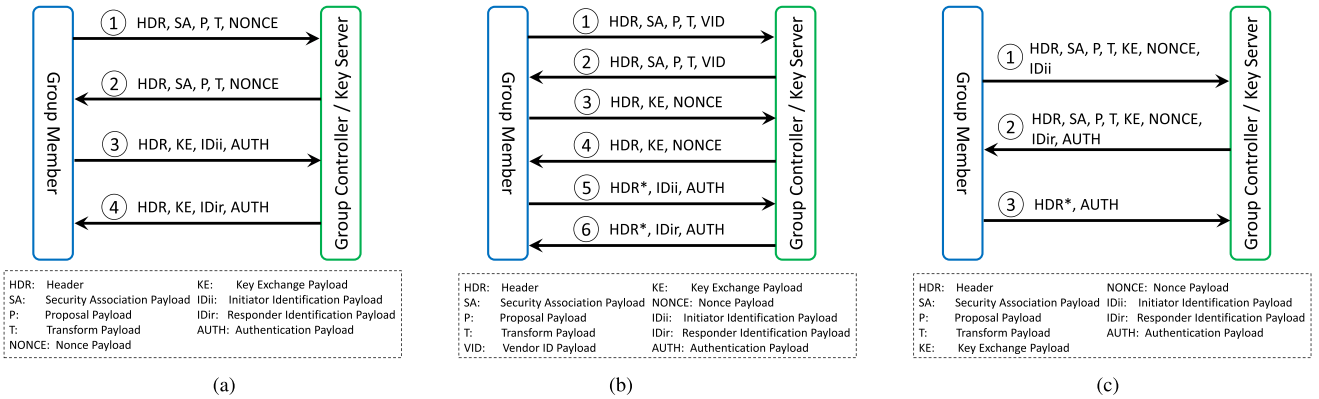


FIGURE 7. GDOI phase-1 authentication based on ISAKMP exchanges. The * indicates that the data is encryption from this point till the end of packet. (a) Base Exchange. (b) Identity Protection Exchange. (c) Aggressive Exchange.

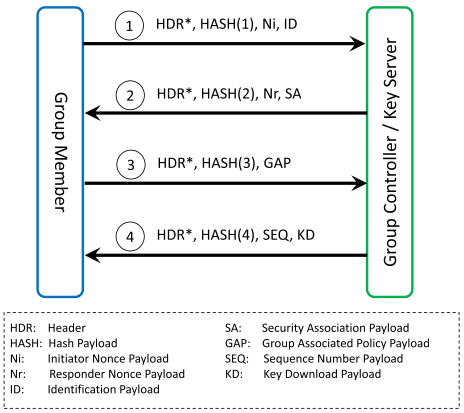


FIGURE 8. GDOI GroupKey-Pull exchange under the protection of phase-1 security credentials. The * indicates that the data is encryption from this point till the end of packet.

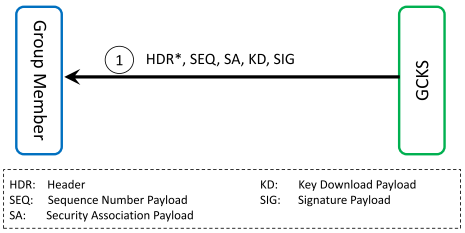


FIGURE 9. GDOI GroupKey-Push exchange under the protection of current KEK security credentials. The * indicates that the data is encryption from this point till the end of packet.

exchange consists of a single message sent by GCKS to GMs as depicted in Fig. 9. It provides updates of KEK, TEK or both. It can be a multicast message to all group members or unicast message to a single group member. Current implementations are using unicast message to all group members upon expiry of previous security credentials. KD payload in Fig. 9 should follow the structure specified in [3] instead of [2]. The signature value in SIG payload is calculated over the entire GroupKey-Push message (except SIG payload content). GroupKey-Push message also contains SEQ payload to achieve protection against replayed packets. It is worth to mention that sequence number in SEQ payload is incremented only when a new GroupKey-Push message is sent (GroupKey-Pull uses current value of sequence number without incrementing).

IV. IMPLEMENTATIONS

The GDOI security mechanism is implemented as a standalone software package and integrated into the security gateway. Thus, this section addresses implementation of a GDOI security package (in Section IV-A), implementation of the security gateway software i.e., without GDOI security (in Section IV-B) and integration of GDOI security package in the security gateway software (in Section IV-C).

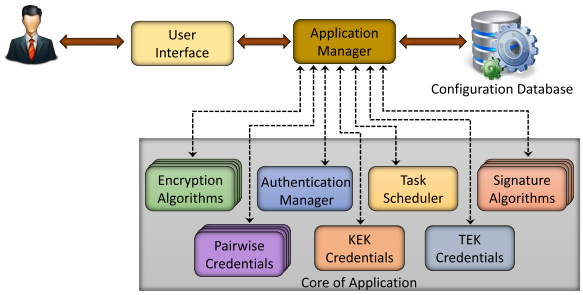


FIGURE 10. Building blocks of GDOI security package.

A. GDOI SECURITY PACKAGE

The GDOI security package is implemented in Linux OS using Python programming language and PyCrypto libraries. The basic reference architecture of the GDOI security package is depicted in Fig. 10. It consists of several important modules/components. The User Interface enables the GDOI application to receive instructions or commands from the user. This module is flexible enough to support multiple user interfaces to allow access to GDOI application both locally (i.e., through command terminal) and over the network (e.g., using TCP or UDP protocols). The Application Manager is

TABLE 1. The configuration database in developed GDOI security package.

	Type	Allowed Values	Description
General	DEVICE_TYPE	GCKS or Member	The application will work either as GCKS or group member based on value of DEVICE_TYPE.
	REMOTE_PEER_ADDRESS	XXX.XXX.XXX.XXX	Destination IPv4 address.
	ISAKMP_TRANSPORT_PROTOCOL	UDP or TCP	Default ISAKMP transport protocol is UDP.
	ISAKMP_TRANSPORT_PORT	500	Default transport port number is 500.
	GDOI_TRANSPORT_PROTOCOL	UDP or TCP	Default GDOI transport protocol is UDP.
	GDOI_TRANSPORT_PORT	848	Default transport port number is 848.
	INTEGRATION_ADDRESS	127.0.0.1	Used for GDOI integration with other protocols e.g., IEC 61850-90-5. Default is loopback interface.
	INTEGRATION_UDP_PORT	3933	Used for GDOI integration with other protocols e.g., IEC 61850-90-5. Default port number is 3933.
	LOG_LEVEL	debug, info, warning or error	Log level: Ignore printing log messages less severe than value specified.
	LOG_COLORED	yes or no	Print debug, info, warning and error messages in different colors.
Authentication	KEY_EXCHANGE_METHOD	Diffie-Hellman	The key exchange/establishment mechanism.
	OAKLEY_GROUP	default-768-bit-MODP	Group in which Diffie Hellman exchange is negotiated.
	LIFE_TYPE	Seconds	Parameter indicating security credentials life (i.e., seconds or kilobytes).
	LIFE_DURATION	86400 or any integer value	The life value of security credentials.
	KEY_LENGTH	128 or 256	The length of key in bits.
	ENCRYPTION_ALGORITHM	AES-256-CBC	Algorithm used for encryption/decryption.
	SIGNATURE_ALGORITHM	HMAC-SHA256	Algorithm used for signature calculation.
Key Encryption Key	LIFE_TYPE	Default or Seconds	Parameter indicating security credentials life (i.e., seconds or kilobytes).
	LIFE_DURATION	Default or any integer value	The life value of security credentials.
	KEY_LENGTH	Default, 128 or 256	The length of key in bits.
	ENCRYPTION_ALGORITHM	Default, AES-128-GCM or AES-256-GCM	Algorithm used for encryption/decryption.
	SIGNATURE_ALGORITHM	Default, HMAC-SHA256-80, HMAC-SHA256-128, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, AES-GMAC64 or AES-GMAC128.	Algorithm used for authentication/signature calculation.
Traffic Encryption Key	LIFE_TYPE	Default or Seconds	Parameter indicating security credentials life (i.e., seconds or kilobytes).
	LIFE_DURATION	Default or any integer value	The life value of security credentials.
	KEY_LENGTH	Default, 128 or 256	The length of key in bits.
	ENCRYPTION_ALGORITHM	Default, AES-128-GCM or AES-256-GCM	Algorithm used for encryption/decryption.
	SIGNATURE_ALGORITHM	Default, HMAC-SHA256-80, HMAC-SHA256-128, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, AES-GMAC64 or AES-GMAC128.	Algorithm used for authentication/signature calculation.

responsible for managing and controlling different GDOI functionalities. It performs coordination activities between different modules in the core of the GDOI application and also enables the user to control the application in real-time. Further, it also updates the user about the current state of the application and any problem that arises during the application run-time.

The Configuration Database stores all the user specified configurations which are loaded at run time. The configurations belong to different modules and control their functionalities. Table 1 lists all configurable options in GDOI security package which are divided into four categories: (i) general configurations, (ii) authentication configurations, (iii) KEK configurations and (iv) TEK configurations. Note that GCKS and GM are implemented as a single software package and DEVICE_TYPE in general configurations decides the role of the application. The INTEGRATION_ADDRESS indicates the socket address on which GDOI GM provides TEK security credentials acquired from GCKS. It is used to integrate GDOI security package into the security gateway software. The authentication configurations are relevant to GDOI GMs (i.e., when DEVICE_TYPE is Member) used during authentication and pairwise key establishment (i.e., GDOI phase-1). Whereas, KEK and TEK configurations are used by GCKS (i.e., when DEVICE_TYPE is GCKS) during GroupKey-Pull and GroupKey-Push exchanges (i.e., GDOI phase-2).

As depicted in Fig. 10, the core of GDOI application consists of seven modules. Encryption Algorithms module contains encryption and decryption methods for different algorithms. At present, the implementations support three encryption algorithms: AES-128-GCM, AES-256-GCM and AES-256-CBC. The choice of encryption algorithm for different GDOI security associations should be specified in configuration database. The Signature Algorithms module contains cryptographic signature calculation and verification methods for different algorithms. At present, the implementations support seven signature algorithms: HMAC-SHA256-80,HMAC-SHA256-128,HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, AES-GMAC64 and AES-GMAC128. The choice of signature algorithm for different GDOI security associations should also be specified in configuration database. The Authentication Manager module implements authentication and pairwise key establishment techniques. At present, Diffie-Hellman is the only supported technique. Task Scheduler module is responsible for executing a task at a given time instant such as keys validity check, or other periodic events. The basic architecture of the Task Scheduler is shown in Fig. 11. Internally, it consists of a queue of pending tasks. Each task has a due time to execute. If the task is a periodic task (e.g., periodic keys replacement), it is placed back into the queue with its new due time. The waiting queue of tasks always remains sorted w.r.t due time. Pairwise Credentials module in Fig. 10 is responsible for managing records of pairwise security policies and keying material between GCKS and each GM. KEK Credentials and TEK Credentials modules are responsible for managing KEK and TEK security policies and keying material, respectively. Since, both KEK and TEK have certain

TABLE 2. The configuration database of the security gateway software.

	Type	Allowed Values	Description
General	GATEWAY_TYPE	Encrypter or Decrypter	Security gateway will perform either encryption or decryption functionalities depending on its location.
	GATEWAY_OUTPUT_FORMAT	IEEE-C37.118 or IEC-61850-90-5	Format of security gateway output packets.
	INTEGRATION_ADDRESS	127.0.0.1	Used for GDOI integration in security gateway. Default is loopback interface.
	INTEGRATION_UDP_PORT	3933	Used for GDOI integration in security gateway. Default port number is 3933.
	PRINT_PACKETS	yes or no	Print input and output packets of security gateway.
	LOG_LEVEL	debug, info, warning or error	Log level: Ignore printing log messages less severe than value specified.
	LOG_COLORED	yes or no	Print debug, info, warning and error messages in different colors.
IEEE C37.118.2	REMOTE_PEER_ADDRESS	XXX.XXX.XXX.XXX	Destination IPv4 address for IEEE C37.118.2.
	SERVER_PORT_TCP	Integer value	TCP transport port used in IEEE C37.118.2 communication. Default is 4712.
	SERVER_PORT_UDP	Integer value	UDP transport port used in IEEE C37.118.2 communication. Default is 4713.
	DATA_MESSAGE_TP	UDP or TCP	Transport protocol to be used for IEEE C37.118.2 data messages.
	CONFIG_MESSAGE_TP	UDP or TCP	Transport protocol to be used for IEEE C37.118.2 configuration messages.
	COMMAND_MESSAGE_TP	UDP or TCP	Transport protocol to be used for IEEE C37.118.2 command messages.
	HEADER_MESSAGE_TP	UDP or TCP	Transport protocol to be used for IEEE C37.118.2 header messages.
IEC 61850-90-5	SOCKET_OPTION	Multicast or Unicast	Send packets in multicast or unicast mode.
	MULTICAST_GROUP	XXX.XXX.XXX.XXX	Address of multicast group in which to listen or sent packets.
	MULTICAST_PORT	Integer value	The UDP port number used for multicast group. Default is 102.
	REMOTE_PEER_ADDRESS	XXX.XXX.XXX.XXX	Destination IPv4 address for IEC 61850-90-5 if SOCKET_OPTION is unicast.
	UDP_PORT	Integer value	The UDP port number if SOCKET_OPTION is unicast. Default is 102.
	PROTOCOL_CHOICE	R-SV or R-GOOSE	IEC 61850-90-5 protocol used for sending packets. Note: packets can be received on both.

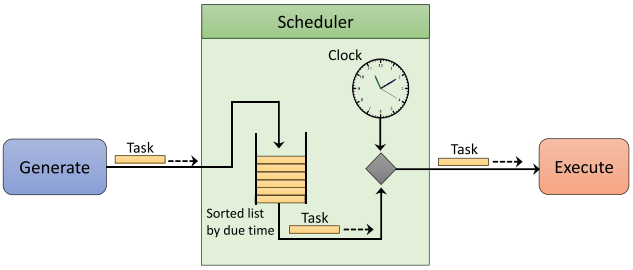


FIGURE 11. The task scheduler module.

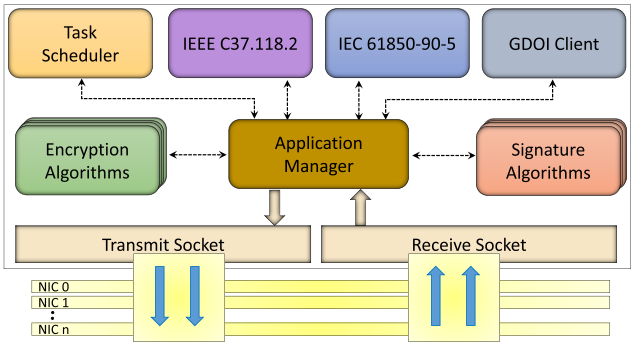


FIGURE 12. The building blocks of the security gateway software.

validity, these modules are also responsible for periodically generating new security credentials and distributing to all authenticated GMs.

B. SECURITY GATEWAY SOFTWARE

Security gateway software is also implemented in Linux OS using Python programming language. Its main building blocks are depicted in Fig. 12. Like the GDOI security package in Fig. 10, the security gateway software also contains User Interface and Configuration Database blocks (not depicted in Fig. 12 for sake of simplicity). The user interface module is flexible and can support multiple local or network based interfaces. Table 2 lists configurable options for the security gateway software which are loaded at run time. The configurations are divided into three different categories: (i) general configurations, (ii) IEEE C37.118.2 configurations and (iii) IEC 61850-90-5 configurations. It can be observed in Fig. 3 that the role of the security gateway software is different depending on its location (e.g., substation or control center). Thus, the general configurations contain an option GATEWAY_TYPE which indicates the role of the security gateway. If the security gateway is used for PMU/PDC (i.e., in substation), GATEWAY_TYPE should be encrypter. In control center, GATEWAY_TYPE for the security gateway should be decrypter. The general configurations also contain an option that indicates the output format of the security gateway (i.e., IEEE C37.118.2 or IEC 61850-90-5). Depending on the GATEWAY_TYPE, the output of security gateway will be secure (i.e., encryption and signature applied) or insecure (i.e., no encryption and no signature applied). The IEEE C37.118.2 configurations provide flexibility to choose different transport protocols for different types of messages. The IEC 61850-90-5 configurations allow the security gateway to work in unicast or multicast fashion. Further, it also allows to choose a protocol for sending synchrophasor packets.

The functionalities of Application Manager, Task Scheduler, Encryption Algorithms and Signature Algorithms modules in Fig. 12 are similar as in GDOI security package addressed in Section IV-A. The IEEE C37.118.2 and IEC 61850-90-5 modules provide complete implementations of IEEE C37.118.2 and IEC 61850-90-5 communication frameworks, respectively. Both modules provide encoding and decoding methods of relevant protocols. Due to lack of open source availability of IEEE C37.118.2 and IEC 61850-90-5 libraries, both IEEE C37.118.2 and IEC 61850-90-5 libraries were first implemented in Python and then integrated into the

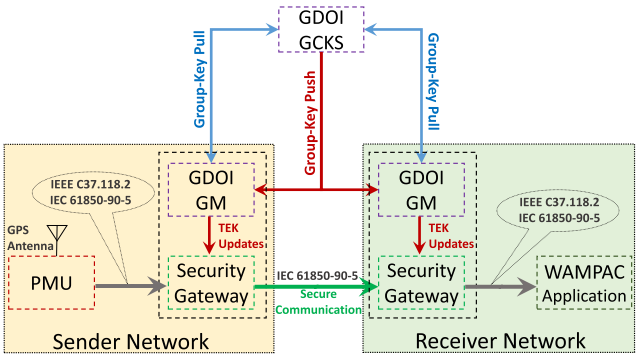


FIGURE 13. The integration of GDOI security package in synchrophasor security gateway. The security gateway can send and receive packets in IEEE C37.118.2 or IEC 61850-90-5 format. The communication between security gateways is based on IEC 61850-90-5 communication framework.

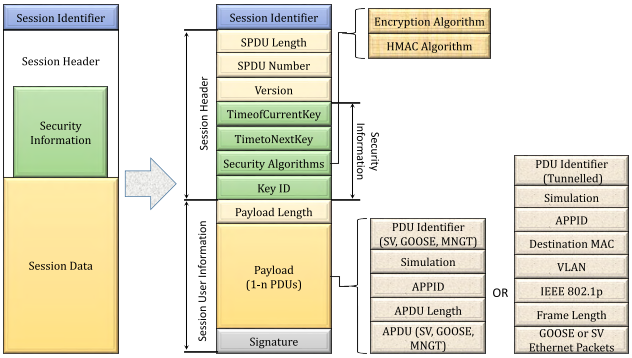


FIGURE 14. Session information inside IEC 61850-90-5 packets.

security gateway software. Security gateway also contains GDOI client. It receives TEK security policies and keying material from GDOI GM over loopback network interface. To decrypt received packets or encrypt transmitting packets, Application Manager acquires security credentials from GDOI client.

C. INTEGRATION OF GDOI SECURITY PACKAGE IN SECURITY GATEWAY

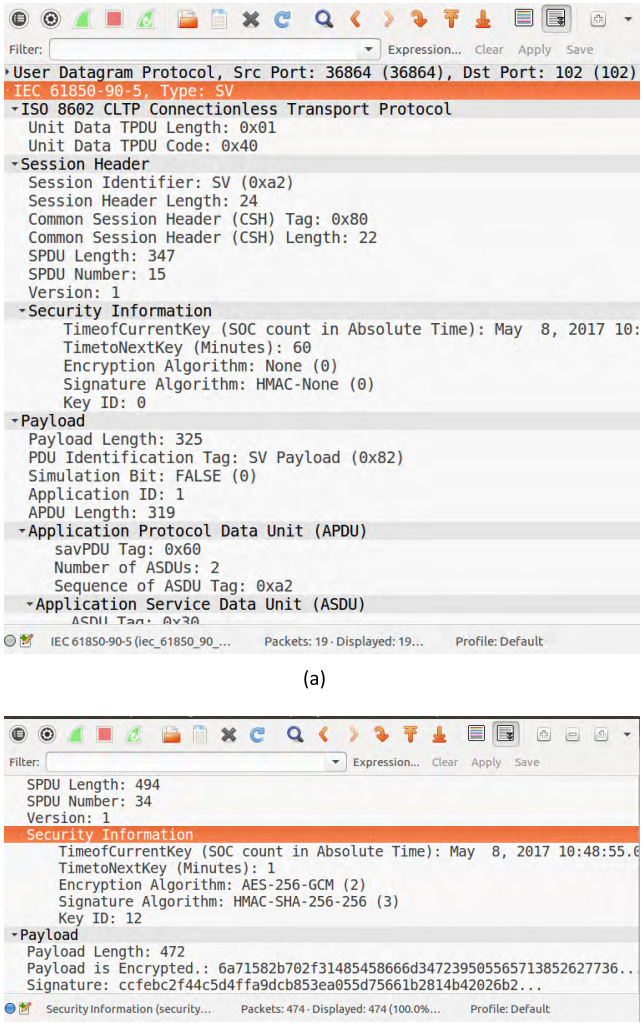
Fig. 13 depicts the integration of two separately implemented software packages i.e., GDOI security package and security gateway. To this aim, a network interface is created on which GDOI GM provides TEK and its associated security policies to the security gateway software. Loopback network interface can be the ideal choice as GDOI GM and security gateway software are expected to run on the same device. However, integration interface is configurable in the configuration database of both software. After acquiring TEK security credentials, security gateway software can successfully encrypt and decrypt packets.

Since security credentials have certain validity and replaced periodically, the security gateway in the receiver network needs to know what security policies and key were used on any received packet. Note: GCKS also replaces security credentials when a GM leaves the group. To this aim, security information is included inside IEC 61850-90-5 session header as depicted in Fig. 14. This includes key ID, IDs of encryption and signature algorithms, TimeofCurrentKey (i.e., time when the key was first assigned by GCKS) and TimetoNextKey (i.e., time when the key will expire). This information helps the security gateway in the receiver network to know if it has the right security credentials available to process the packet. If security information mismatches, it forwards the original packet to WAMPAC application without processing (i.e., a possible scenario if PMU itself has applied its own security credentials on packet).

V. EXPERIMENTAL EVALUATION

The testbed used for experimental evaluation is similar to Fig. 13. It consists of five devices with different functionalities: (i) PMU/PDC, (ii) Encrypter Gateway (i.e., security gateway plus GDOI GM in sender network), (iii) Decrypter Gateway (i.e., security gateway plus GDOI GM in receiver network), (iv) WAMPAC application and (v) GDOI GCKS. To reduce the complexity and cost of synchrophasor-based systems, encrypter gateway, decrypter gateway and GDOI GCKS were executed on compact and inexpensive ARM based device i.e., Raspberry Pi v2 (ARM Cortex-A7 CPU 900 MHz, RAM 1 GB). The PMU was a commercial PMU from ABB that supports only IEEE C37.118.2 communication framework and was configured to provide three phase measurements. However, to evaluate the security gateway for PDC and PMU with IEC 61850-90-5 support, ePMU and ePDC were implemented using developed IEEE C37.118.2 and IEC 61850-90-5 libraries. The ePMU and ePDC were executed on a standard PC and were very flexible and configurable. The WAMPAC application was executed on a standard PC and supports both, IEEE C37.118.2 and IEC 61850-90-5 communication frameworks. For sake of experiments, it was designed to provide only monitoring functionalities by properly decoding and visualizing received synchrophasor data.

The first step in experimental evaluation is to verify the correct functioning of all developed software entities including IEEE C37.118.2 libraries, IEC 61850-90-5 libraries, ePMU, ePDC, security gateway, GDOI security package and its integration in the security gateway. The functional verification of IEEE C37.118.2 libraries and IEC 61850-90-5 libraries was performed using the PMU Connection Tester tool. The PMU Connection Tester successfully decoded and visualized synchrophasor data indicating the correctness of implementations. The functional evaluation of ePMU and ePDC was performed by analyzing their output with Wireshark packet capturing tool. It is notable that Wireshark lacks support for IEC 61850-90-5 communication framework at present. Thus, a dissector, compliant with the specifications described in IEC 61850-90-5 technical document [3], was implemented and integrated in Wireshark (as shown in Fig. 15). Under different configurations of ePMU and ePDC (e.g., output in IEEE C37.118.2 or IEC 61850-90-5 format, PDC with specified number of PMUs data, etc), their communication with



(a)

(b)

FIGURE 15. Functional analysis of R-SV traffic using the implemented IEC 61850-90-5 compliant Wireshark dissector. (a) IEC 61850-90-5 R-SV without GDOI. (b) IEC 61850-90-5 R-SV with GDOI.

the security gateway was analyzed in Wireshark. Analyzing configuration and data messages of IEEE C37.118.2 and R-SV messages of IEC 61850-90-5 in Wireshark verified correct functioning of ePMU and ePDC emulators. Functional verification of GDOI security package was also performed and GM was able to successfully authenticate with GCKS and received updated security policies and keying material. GM provided security credentials to the security gateway on loopback network interface which were then used for encryption/decryption and cryptographic signature calculation/verification in synchrophasor packets. Analyzing network traffic between encrypter gateway and decrypter gateway in Wireshark verified that security credentials were successfully updated on periodic basis (depending on GCKS configurations). The encrypter gateway was able to parse and convert received IEEE C37.118.2 or insecure IEC 61850-90-5 packets into secure IEC 61850-90-5 packets. Analyzing the output of the decrypter gateway in Wireshark verified that secure IEC 61850-90-5 packets were successfully converted back into original IEEE C37.118.2 or insecure IEC 61850-90-5 packets depending on the configurations of decrypter gateway.

After functional verification of all developed software entities in experimental testbeds, the next step is to measure performance critical factors (e.g., communication overhead, latencies, resource requirements, etc) under extreme/worse conditions which may impact the effectiveness or real-time performance of security gateway. The main objective of measuring performance metrics is to investigate if the compact low-cost ARM-based Raspberry Pi is suitable to function as security gateway when deployed in legacy synchrophasor-based networks. Also performance metrics help to analyze potential impact of security gateway and especially the GDOI security mechanism on different types of synchrophasor applications with very strict data rate and latency requirements. Depending on the configurations, the computational complexity and resource requirements for GDOI security mechanism might be relatively low. However, the security gateway faces challenges to meet the requirements of latencies and data transmission rates. Thus, the quantitative and qualitative evaluation is divided into two parts: (i) performance evaluation of GDOI security package (presented in Section V-A) and (ii) performance evaluation of security gateway (presented in Section V-B).

A. PERFORMANCE EVALUATION OF GDOI SECURITY PACKAGE

This section evaluates GDOI security package and presents results related to GCKS and GM as shown in the testbed in Fig. 13. It mainly analyzes network traffic overhead in GDOI security mechanism, required network resources and computational cost for GCKS.

1) NETWORK TRAFFIC OVERHEAD

Network traffic overhead is considered a significant performance metric for any security mechanism. It affects the maximum size of real-information/data that can be carried inside a single packet. However, for security mechanism based on Key Distribution Center (KDC), network traffic overhead information is useful to determine the required communication channel capacity enough for transmitting certain number of messages. The KDC/GCKS might be authenticating and providing security policies and keying material to potentially hundreds or thousands of client devices. Thus, the communication channel should have enough capacity to allow smooth and reliable communication between GCKS and its all GMs.

Fig. 16 analyzes network traffic overhead for GDOI security mechanism. During the experiment, a GM authenticates with GCKS using IPE exchange, performs GroupKey-Pull and GroupKey-Push exchanges, intentionally sends an error notification message and finally instructs GCKS to remove it from the group and delete its already established security association. Fig. 16(a) depicts total Bytes exchanged

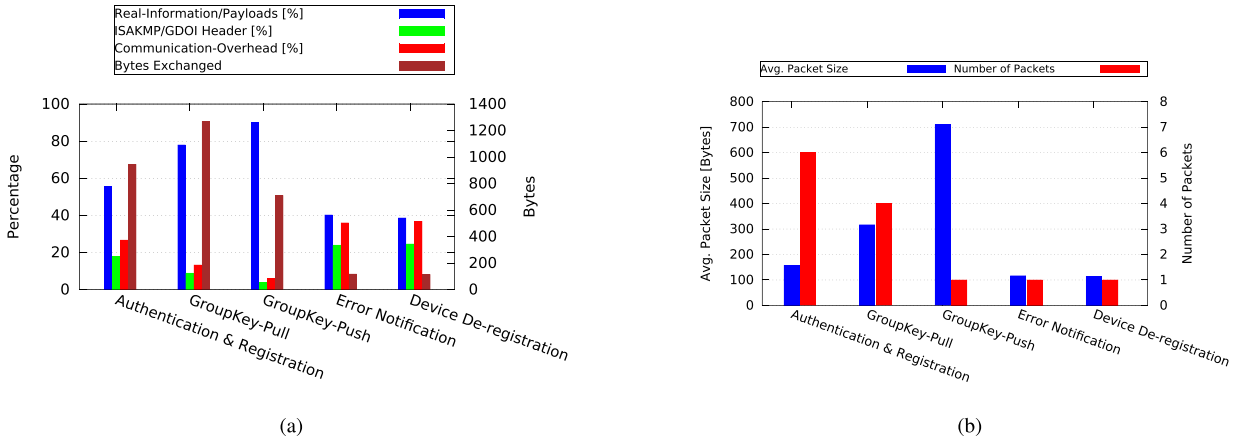


FIGURE 16. Analysis of network traffic overhead for GDOI security mechanism. (a) Percentage overhead and total Bytes exchanged during different GDOI phases. (b) Average packet size and total number of packets exchanged during different GDOI phases.

during each individual GDOI phase. It also classifies packets and presents the percentage of real-information (i.e., actuals payloads), percentage of overhead Bytes due to ISAKMP/GDOI header and percentage of overhead Bytes due to communication (i.e., transport, network and Ethernet layers). It can be observed that amount of real-information is quite high compared to communication overhead during authentication, GroupKey-Pull and GroupKey-Push. However, packets during error notification and device de-registration are small and contain high percentage of overhead Bytes. For better understanding of the network traffic overhead, Fig. 16(b) depicts the average packet size and total number of packets exchanged during each individual GDOI phase. It can be observed in Fig. 16 that certain packets are quite big and contain significant overhead Bytes as well. However, all the packets are exchanged only once for each GM except the GroupKey-Push (i.e., depends on GCKS configurations and contains periodic updates for KEK and TEK) and error notification (i.e., only sent in case of error in decoding a received packet).

2) RESOURCE REQUIREMENTS

Based on the fact that GCKS is executed on the Raspberry Pi with limited memory, it should be able to provide service to hundreds or thousands of client devices (i.e., GMs). To assess memory requirement, increasing number of emulated client devices were registered with GCKS. It can be observed in Fig. 17 that each additional client device requires approximately 3.51 KB of memory at GCKS. Due to very low memory requirement, the 1 GB available memory is more than enough to easily offer security services to thousands of client devices. Furthermore, we observed that CPU usage was quite low during client device authentication and GroupKey-Pull. During steady state (i.e., when no new device authentication is performed), CPU usage was negligible as GCKS does not perform any task/computations. Note that GroupKey-Push messages are very infrequent.

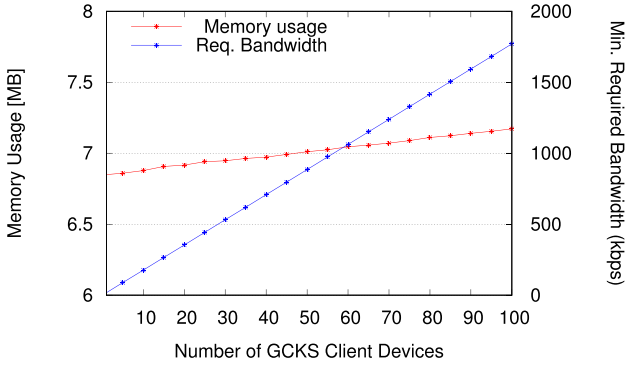


FIGURE 17. Memory and bandwidth requirement with increasing number of registered GCKS client devices. Note: bandwidth requirement considers simultaneous registration from client devices (i.e., Bytes exchange during GDOI phase-1 authentication and GroupKey-Pull).

To avoid traffic congestion and packet loss, minimum required bandwidth/channel capacity should be available for GCKS. Fig. 17 also depicts communication channel bandwidth requirement when simultaneous authentication and GroupKey-Pull requests are received from increasing number of client devices. It can be observed that bandwidth requirement is less than 1.8 Mbps even when 100 client devices (well beyond to expect in realistic scenarios) simultaneously perform authentication with GCKS. This makes GCKS suitable for most available access technologies today (e.g., ADSL lite, ADSL2+, etc).

3) LATENCIES ANALYSIS

Another performance critical factor for real-time applications is processing latency. The GCKS processing latencies indirectly represent how many client devices authentication/GroupKey-Pull can be performed per second. Comparatively, high latencies are considered more critical for GroupKey-Push that provides security policies and keying material updates to authenticated clients. Long delays in receiving updated security credentials may leave client device

TABLE 3. Processing latencies during different phases of GDOI.

GDOI Phase	Processing Latencies (ms)			
	Min	Avg	Max	Std. Dev
Authentication	106.83	129.76	154.52	3.01
GroupKey-Pull	85.79	110.24	123.53	2.79
GroupKey-Push	16.14	21.96	24.78	1.31

with out-dated security credentials and unable to communicate with other GMs. Table 3 presents latencies experienced during authentication, GroupKey-Pull and GroupKey-Push exchanges. The latencies are averaged over 100 trials and include both, GCKS and GM processing latencies. It can be observed in Table 3 that latencies are quite small. Excluding GM processing latencies, GCKS can easily perform authentication and GroupKey-Pull with up to 5 client devices per second. It is still a quite good number considering low processing power of GCKS (i.e., Raspberry Pi). With the configuration of GCKS to provide new security policies and keying material to client devices 5 seconds before the expiry of old security credentials, the 21.96 ms average GroupKey-Push latency does not severely impact the GDOI operations.

B. PERFORMANCE EVALUATION OF SECURITY GATEWAY

This section evaluates the effectiveness of security gateway in both sender (i.e., encrypter gateway) and receiver (i.e., decrypter gateway) networks as shown in the testbed in Fig. 13. It mainly analyzes the impact of security gateway on required network resources and evaluates its suitability for different types of synchrophasor applications.

For experiments reported in this section, ePMU is configured to include three phasor integer values in polar format, two analog values in floating point format and one digital word (i.e., representing 16 digital status points) in each IEEE C37.118.2 data message. Note frequency and Rate Of Change Of Frequency (ROCOF) are also in integer format. These configurations are reported by ePMU to the security gateway in IEEE C37.118.2 configuration message. Same settings are used when ePMU uses IEC 61850-90-5 R-SV. In case of ePDC, all the embedded PMUs have same configurations as the ePMU.

1) NETWORK TRAFFIC OVERHEAD AND BANDWIDTH REQUIREMENT

Network traffic overhead is a very critical factor for synchrophasor-based real-time control and monitoring in smart grid. High data transmission rates of bulky packets containing significant overhead Bytes can increase the requirement of channel capacity/bandwidth. On low bandwidth channels, throughput is normally reduced and probability of packet loss increases. Synchrophasor-based real-time control and monitoring applications normally involve high data transmission rates and require communication channels with enough transmission capacity. Depending on the synchrophasor application, any packet loss could result in devastating consequences.

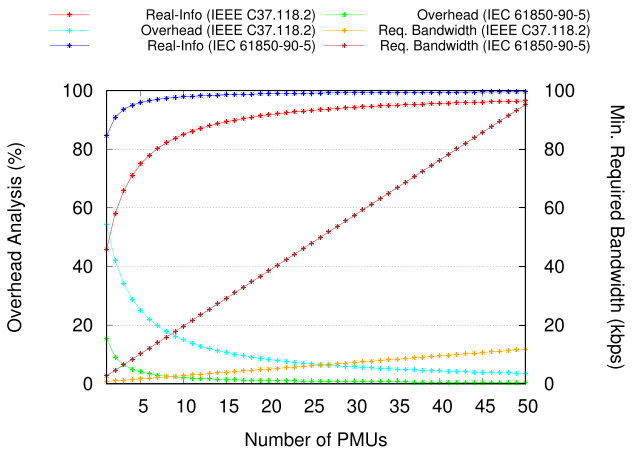


FIGURE 18. Analysis of network traffic overhead and bandwidth requirement for PDC aggregating data from increasing number of PMUs.

To analyze the impact of the security gateway on network traffic overhead, Fig. 18 depicts overhead for IEEE C37.118.2 data messages and IEC 61850-90-5 R-SV messages for ePDC aggregating data from increasing number of PMUs. Real-information in Fig. 18 designates the actual synchrophasor data inside packets while overhead represents the additional Bytes added to packets (at link, network and transport layers). With increase in the number of PMUs data inside PDC, the percentage of real-information inside packets increases and the overhead percentage decreases. Fig. 18 also depicts bandwidth requirement when a single IEEE C37.118.2 or IEC 61850-90-5 packet is transmitted per second. It can be observed that IEEE C37.118.2 has comparatively much higher overhead percentage but its bandwidth requirement is much lower than IEC 61850-90-5. It is due to the fact that IEEE C37.118.2 packets are much smaller in size than IEC 61850-90-5 packets for carrying the same amount of information.

The bandwidth requirement depends on: (i) PDC types (i.e., how many PMUs data is aggregated?), (ii) communication framework (IEEE C37.118.2 or IEC 61850-90-5), and (iii) data transmission rate. Fig. 18 depicts bandwidth requirement when a single packet is transmitted per second. Synchrophasor applications normally involve very high data transmission rates. Authors in [24] classified synchrophasor applications into 5 different categories depending on the required message transmission rates: Very Low (1 packet/second), Low (up to 30 packets/second), Medium (up to 60 packets/second), High (up to 120 packets/second), and Ultra (up to 720 packet/second). In Table 4, the bandwidth requirement has been analyzed for these five category of synchrophasor applications. It can be observed in Table 4 that the use of security gateway (whose output is in IEC 61850-90-5 format) will significantly increase the channel bandwidth requirement as most commercial PMUs support IEEE C37.118.2. The difference in the bandwidth requirement

TABLE 4. Bandwidth requirement for IEEE C37.118.2 and IEC 61850-90-5 synchrophasor communication frameworks.

Device Type	Synchrophasor Data Reporting Rate				
	Very Low (1 pps)	Low (30 pps)	Medium (60 pps)	High (120 pps)	Ultra (720 pps)
PMU	0.77 kbps / 2.69 kbps	23.04 kbps / 80.88 kbps	46.08 kbps / 161.76 kbps	92.16 kbps / 323.52 kbps	552.96 kbps / 1.94 Mbps
PDC (5 PMUs)	1.66 kbps / 10.25 kbps	49.92 kbps / 307.44 kbps	99.84 kbps / 0.61 Mbps	199.68 kbps / 1.23 Mbps	1.19 Mbps / 7.38 Mbps
PDC (10 PMUs)	2.78 kbps / 19.68 kbps	83.52 kbps / 590.64 kbps	167.04 kbps / 1.18 Mbps	334.08 kbps / 2.36 Mbps	2.0 Mbps / 14.17 Mbps
PDC (25 PMUs)	6.14 kbps / 48.0 kbps	184.32 kbps / 1.44 Mbps	368.64 kbps / 2.88 Mbps	737.28 kbps / 5.76 Mbps	4.42 Mbps / 34.56 Mbps
PDC (50 PMUs)	11.74 kbps / 95.21 kbps	352.32 kbps / 2.86 Mbps	704.64 kbps / 5.71 Mbps	1.41 Mbps / 11.42 Mbps	8.45 Mbps / 68.55 Mbps

Values to the left of '/' are for IEEE C37.118.2 and to the right are for IEC 61850-90-5.

TABLE 5. Memory requirement for security gateway.

GDOI Group Member	Security Gateway	Total Requirement
6.873 MB	13.658 MB	20.53 MB

for IEEE C37.118.2 and IEC 61850-90-5 significantly increases with increase in the data transmission rate and PDC type. For very high data transmission rates and PDC (i.e., aggregating data from 50 PMUs), many access technologies such as ADSL lite(downstream 1.5 Mbps and upstream 0.5 Mbps), ADSL1 (downstream 8 Mbps and upstream 1.0 Mbps), ADSL2 (downstream 12 Mbps and upstream 1.3 Mbps), ADSL2+ (downstream 24 Mbps and upstream 1.4 Mbps), ADSL2+M (downstream 24 Mbps and upstream 3.3 Mbps), etc could not provide enough capacity. Optical fiber network (which supports over 100 Gbps) could be a suitable choice. However, not every country has optical fiber network installed. It is worth to mention that this is not the limitation of security gateway but the IEC 61850-90-5 communication framework. If PMU/PDC provides data to security gateway in IEC 61850-90-5 format, the security gateway will have negligible impact on bandwidth requirement. Due to limitations of IEEE C37.118.2 and unique/useful features of IEC 61850-90-5 [1], IEC 61850-90-5 will sooner or later become part of the synchrophasor systems.

2) MEMORY REQUIREMENT

Table 5 presents memory requirement for security gateway based on the experimental testbed depicted in Fig. 13. Total memory requirement should take into account memory required by GCKS GM as well as memory required for security gateway software. It is obvious in Table 5 that total memory requirement is very low and even traditional gateway devices (which are normally equipped with 32 or 64 MB memory) can meet the requirement for synchrophasor's security gateway.

3) COMPUTATIONAL COMPLEXITY

Security gateway software is intended to be executed on low power Raspberry Pi with limited processing capabilities, the computational complexity could impact its suitability for different types of synchrophasor applications. Security gateway processing latency is a very critical factor

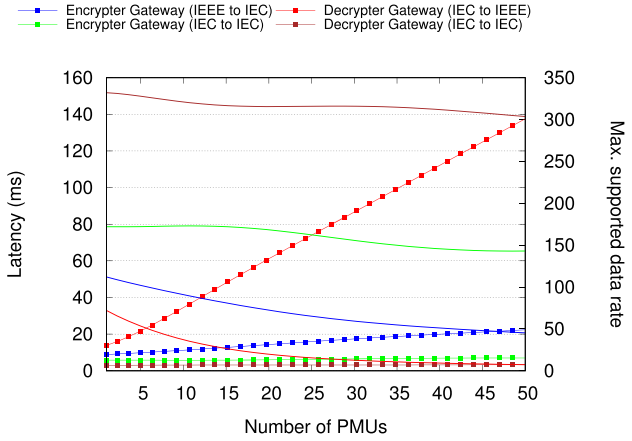


FIGURE 19. Security gateway processing latency and maximum supported data rate when no encryption and no signature is used (i.e., security gateway transmits insecure IEC 61850-90-5 packets).

that could limit the throughput or maximum supported data rate. High latency means low supported data rate by the security gateway. The computational complexity for the encrypter gateway (i.e., security gateway in sender network) could be different from the decrypter gateway (i.e., security gateway in receiver network) due to different functionalities involved. The final supported data rate depends on the security gateway (encrypter or decrypter) with maximum latency value.

Security gateway processing latencies depend on two factors: (i) conversion of packets from one communication framework into the other, and (ii) complexity of security algorithms (i.e., encryption and cryptographic signature). To assess the computational complexity in securing incoming packets, the security gateway processing latencies were measured while it is receiving IEEE C37.118.2 data messages and IEC 61850-90-5 R-SV messages from ePDC that aggregates data from increasing number of PMUs. Fig. 19 depicts processing latencies and maximum supported data rates when the security gateway does not apply any encryption and signature calculation/verification. In this case, processing latencies solely depend on the computational complexity involved in converting IEEE C37.118.2 packets into IEC 61850-90-5 packets and vice versa. It can be observed in Fig. 19 that processing latencies are high when input of encrypter gateway and output of decrypter gateway are in IEEE C37.118.2 format. Further, processing latencies increase rapidly when a PDC carries data from a large number of PMUs.

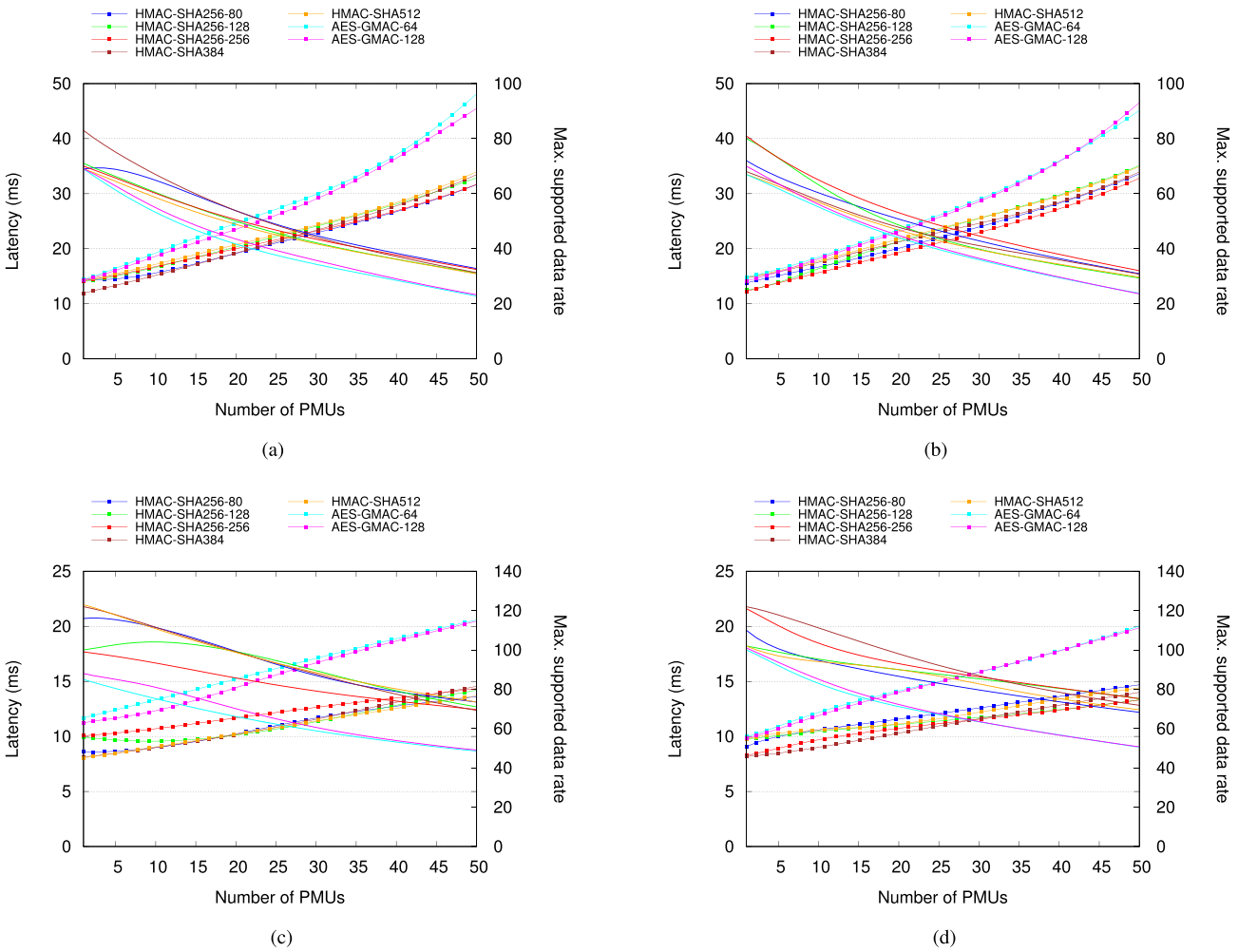


FIGURE 20. Encrypter gateway processing latency and maximum supported data rate with increasing number of PMUs data inside packets. Dotted lines represent latencies while smooth lines represent data rates. Results are averaged over 100 trials. (a) IEEE to IEC using AES-128-GCM. (b) IEEE to IEC using AES-256-GCM. (c) IEC to IEC using AES-128-GCM. (d) IEC to IEC using AES-256-GCM.

TABLE 6. Analysis of the security gateway suitability for different types of synchrophasor applications.

	Application Criticality	Latency (ms)	Rate (Hz)	Encrypter Gateway	Decrypter Gateway	Conclusion
IEEE C37.118.2	Ultra	5-20	120-720+	○ ○ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○
	High	20-50	60-120	◑ ◑ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○
	Medium	50-100	30-60	● ● ● ◑ ◑	◑ ◑ ○ ○ ○	◑ ◑ ○ ○ ○
	Low	100-1000	1-30	● ● ● ● ●	● ● ● ◑ ◑	● ● ● ◑ ◑
	Very Low	>1000	<1	● ● ● ● ●	● ● ● ● ●	● ● ● ● ●
IEC 61850-90-5	Ultra	5-20	120-720+	○ ○ ○ ○ ○	○ ○ ○ ○ ○	○ ○ ○ ○ ○
	High	20-50	60-120	● ◑ ◑ ◑ ◑	◑ ◑ ◑ ◑ ◑	◑ ◑ ◑ ◑ ◑
	Medium	50-100	30-60	● ● ● ● ●	● ● ● ● ●	● ● ● ● ●
	Low	100-1000	1-30	● ● ● ● ●	● ● ● ● ●	● ● ● ● ●
	Very Low	>1000	<1	● ● ● ● ●	● ● ● ● ●	● ● ● ● ●

Black circle = PMU, orange circle = PDC (5 PMUs data), brown circle = PDC (10 PMUs data), blue circle = PDC (15 PMUs data), red circle = PDC (20 PMUs data).
Empty circle means 'Not Supported'. Filled circle means 'Supported'. Half-filled circle means low data rates supported but upper data rate bound not supported.
IEEE C37.118.2 means that the input of encrypter gateway and output of decrypter gateway are in IEEE C37.118.2 format.
IEC 61850-90-5 means that the input of encrypter gateway and output of decrypter gateway are in IEC 61850-90-5 - R-SV format.

To measure the impact of encryption and signature algorithms on the supported data rates by the security gateway, Fig. 20 and Fig. 21 depict processing latencies of encrypter and decrypter gateways, respectively. It can be observed that the difference in computational complexity of AES-128-GCM and AES-256-GCM encryption algorithms

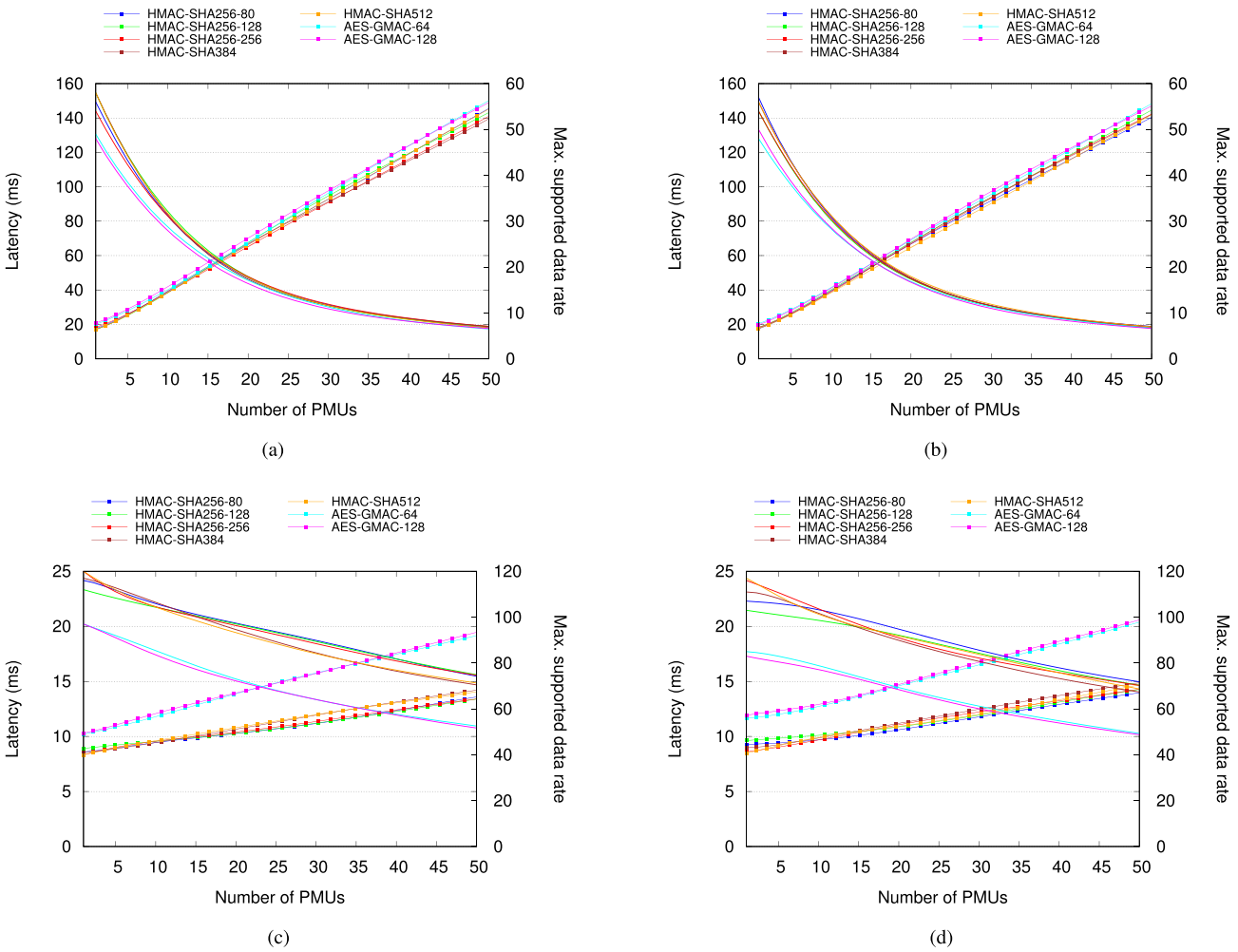


FIGURE 21. Decrypter gateway processing latency and maximum supported data rate with increasing number of PMUs data inside packets. Dotted lines represent latencies while smooth lines represent data rates. Results are averaged over 100 trials. (a) IEC to IEEE using AES-128-GCM. (b) IEC to IEEE using AES-256-GCM. (c) IEC to IEC using AES-128-GCM. (d) IEC to IEC using AES-256-GCM.

is negligible. However, AES-GMAC based signature algorithms have significant higher processing latencies compared to HMAC-SHA based algorithms especially for PDC that aggregates data from large number of PMUs. It can be observed in Fig. 20 and Fig. 21 that the maximum supported data rates by the security gateway depend on what encryption and signature algorithms used. Further, decrypter gateway latencies are slighter higher than the encrypter gateway. Thus, the decrypter gateway supports slightly lower data rates compared to encrypter gateway. It can also be observed that the security gateway supported data rate is low when input of encrypter gateway and output of decrypter gateway are in IEEE C37.118.2 format.

Table 6 analyzes the suitability of the security gateway for different types of synchrophasor applications based on its supported data rates. It considers five categories of applications defined in [24] with strict latency and data rate requirements. The rate in Table 6 represents the number of packets transmitted per second. It can be observed that for PMU, security gateway meets the requirements of very low, low, medium and highly critical synchrophasor applications. For PDC, the security gateway meets the requirements for synchrophasor applications with medium data transmission rates. It is worth noting that security gateway receiving high data rate packets than its supported capacity will result in packet loss. However, results in Table 6 are presented for a Raspberry Pi based security gateway that has very low processing power. For ultra critical synchrophasor applications, the security gateway software should be hosted on a more powerful device.

VI. CONCLUSION

The adoption of secure IEC 61850-90-5 for synchrophasor communication in power grids is a big challenge due to interoperability issues with legacy phasor devices. Most legacy phasor devices use IEEE C37.118.2 which is highly vulnerable to cyber attacks. Firouzi *et al.* [7] recently proposed a solution using an IEEE C37.118.2 to IEC 61850-90-5 gateway. However, the published gateway

has limited features to fully address interoperability issues and also lacks the IEC recommended GDOI security mechanism. Further, previous work [3] lacks a clear functional specification towards practically implementing GDOI security mechanism.

To address interoperability, integration as well as communication security issues for synchrophasor-based systems, this paper presented the design and implementation of a low cost security gateway. The security gateway supports two way conversion features between IEEE C37.118.2 and IEC 61850-90-5 communication frameworks. It is highly configurable and can be used for different types of PMUs and PDCs. Further, it ensures communication security using IEC recommended GDOI security mechanism. The GDOI provides dynamic security policies and keying material by periodically replacing old security credentials. This paper provided clear functional and technical details of GDOI security mechanism which can be used as proof of concept for implementing it in future phasor devices. Further, the GDOI implementation reported in this paper has been adopted in OpenPMU [12].

This paper also functionally validated security gateway features by implementing a Wireshark dissector and performed compliance testing of IEEE C37.118.2 and IEC 61850-90-5 with the PMU Connection Tester. Through detailed experimental evaluation, this paper validated the suitability of GDOI security mechanism for constraint devices such as ARM processor based security gateway. It was observed that security gateway increases the network bandwidth requirement for synchrophasor communication due to using IEC 61850-90-5. However, on most existing access technologies today, it can easily support PMUs and PDCs with data transmission rate of up to 720 packets per second (much higher rate than used in most practical synchrophasor applications). This paper also identified that the processing latency of security gateway is a critical performance limiting factor for synchrophasor applications with strict latency and data rate requirements. It is experimentally validated using different PMUs and PDCs that an effective low cost and compact implementation of the security gateway on ARM processor is feasible for most synchrophasor applications requiring data rate up to 100 packets/second. For applications requiring very high data transmission rates, the security gateway could be deployed on a more powerful device. The result of the presented comprehensive implementation is to enable new smart grid control applications that depend on secure and trustworthy real-time synchrophasor data.

REFERENCES

[1] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "Analysis of IEEE C37.118 and IEC 61850-90-5 synchrophasor communication frameworks," in *Proc. IEEE Power Energy Soc. General Meet. (PESGM)*, Jul. 2016, pp. 1–5.

[2] B. Weis, S. Rowles, and T. Hardjono, *The Group Domain of Interpretation*, document RFC 6407, Internet Engineering Task Force, Oct. 2011.

[3] *Communication Networks and Systems for Power Utility Automation*, document IEC 61850-90-5, 2012.

[4] Y. Wang, T. T. Gamage, and C. H. Hauser, "Security implications of transport layer protocols in power grid synchrophasor data communication," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 807–816, Mar. 2016.

[5] L. Coppolino, S. D'Antonio, and L. Romano, "Exposing vulnerabilities in electric power grids: An experimental approach," *Int. J. Crit. Infrastruct. Protect.*, vol. 7, no. 1, pp. 51–60, 2014.

[6] R. Khan, P. Maynard, K. McLaughlin, D. Laverty, and S. Sezer, "Threat analysis of BlackEnergy malware for synchrophasor based real-time control and monitoring in smart grid," in *Proc. 4th Int. Symp. ICS SCADA Cyber Secur. Res. (ICS-CSR)*, Aug. 2016, pp. 53–63.

[7] S. R. Firouzi, L. Vanfretti, A. Ruiz-Alvarez, H. Hooshyar, and F. Mahmood, "Interpreting and implementing IEC 61850-90-5 routed-sampled value and routed-GOOSE protocols for IEEE C37.118.2 compliant wide-area synchrophasor data transfer," in *Electr. Power Syst. Res.*, vol. 144, pp. 255–267, Mar. 2017.

[8] I. M. Dragomir and S. S. Iliescu, "Synchrophasors applications in power system monitoring, protection and control," in *Proc. 20th Int. Conf. Control Syst. Comput. Sci.*, May 2015, pp. 978–983.

[9] M. Kanabar, M. G. Adamiak, and J. Rodrigues, "Optimizing wide area measurement system architectures with advancements in phasor data concentrators (PDCs)," in *Proc. IEEE Power Energy Soc. General Meet. (PESGM)*, Jul. 2013, pp. 1–5.

[10] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "IEEE C37.118-2 synchrophasor communication framework: Overview, cyber vulnerabilities analysis and performance evaluation," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2016, pp. 1–10.

[11] V. Madani, S. Picard, Y. Yin, and M. Adamiak, "Challenges and lessons learned from commissioning an IEC 61850-90-5 based synchrophasor system," in *Proc. 68th Annu. Conf. Protective Relay Eng.*, 2015, pp. 842–849.

[12] D. M. Laverty, L. Vanfretti, I. A. Khatib, V. K. Applegreen, R. J. Best, and D. J. Morrow, "The OpenPMU project: Challenges and perspectives," in *Proc. IEEE Power Energy Soc. General Meet. (PESGM)*, Jul. 2013, pp. 1–5.

[13] T. Zseby and J. Fabini, "Security challenges for wide area monitoring in smart grids," *e i Elektrotechnik Informationstechnik*, vol. 131, no. 3, pp. 105–111, 2014.

[14] T. Morris *et al.*, "Cybersecurity testing of substation phasor measurement units and phasor data concentrators," in *Proc. ACM Annu. Workshop Cyber Secur. Inf. Intell. Res.*, 2011, p. 24.

[15] J. Stewart *et al.*, "Synchrophasor security practices," in *Proc. 14th Annu. Georgia Tech Fault Disturbance Anal. Conf.*, 2011, pp. 1–10.

[16] S. Pal, B. Sikdar, and J. Chow, "Real-time detection of packet drop attacks on synchrophasor data," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Nov. 2014, pp. 896–901.

[17] S. Paudel, P. Smith, and T. Zseby, "Data integrity attacks in smart grid wide area monitoring," in *Proc. 4th Int. Symp. ICS SCADA Cyber Secur. Res. (ICS-CSR)*, Aug. 2016, pp. 74–83.

[18] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 998–1010, 4th Quart., 2012.

[19] C. Beasley, X. Zhong, J. Deng, R. Brooks, and G. K. Venayagamoorthy, "A survey of electric power synchrophasor network cyber security," in *Proc. PES ISGT-Europe*, Oct. 2014, pp. 1–5.

[20] I. Ali, S. M. S. Hussain, and A. Aftab, "Communication modeling of phasor measurement unit based on IEC 61850-90-5," in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2015, pp. 1–6.

[21] D. Harkins and D. Carrel, *The Internet Key Exchange (IKE)*, document RFC 2409, Internet Engineering Task Force, 1998

[22] D. Piper, *The Internet IP Security Domain of Interpretation for ISAKMP*, document RFC 2407, Internet Engineering Task Force, 1998.

[23] D. Maughan, M. Schertler, M. Schneider, and J. Turner, *Internet Security Association and Key Management Protocol (ISAKMP)*, document RFC 2408, Internet Engineering Task Force, 1998.

[24] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead, and G. C. Zweigle, "Smart generation and transmission with coherent, real-time data," *Proc. IEEE*, vol. 99, no. 6, pp. 928–951, Jun. 2011.

RAFIULLAH KHAN received the B.Sc. degree in electrical engineering from the University of Engineering and Technology Peshawar, Pakistan, in 2009, the master's degree in satellite navigation and related applications from the Politecnico Di Torino, Italy, in 2010, and the Ph.D. degree jointly from the University of Genoa, Italy, and the Polytechnic University of Catalonia-BarcelonaTech, Spain, in 2014. His Ph.D. degree was supported by the Erasmus Mundus Program, funded by the European Commission.

He is currently a Post-Doctoral Research Fellow with Queen's University Belfast, U.K. He has co-authored over 35 scientific publications in international journals and conference proceedings and carried our research activities in framework of several national and European research projects. His research interests include ad hoc networking, cyber security, cyber physical systems, and critical infrastructure protection.

KIERAN MCLAUGHLIN received the M.Eng. in electrical and electronic engineering and the Ph.D. degree from Queen's University Belfast, U.K., in 2003 and 2006, respectively. He is currently a Lecturer with the Center for Secure Information Technologies, where he leads research in cyber security for smart grids, industrial control systems, and supervisory control and data acquisition networks. His research interests include threat analysis, intrusion detection/prevention, and cyber-physical resilience measures.

DAVID LAVERTY received the M.Eng. and Ph.D. degrees from Queen's University Belfast, Belfast, U.K., in 2006 and 2010, respectively. He is currently a Lecturer with the Energy, Power and Intelligent Control Cluster, Queen's University Belfast, Belfast, U.K. His current research interests include anti-islanding detection, cyber-security and telecommunications, and synchrophasor measurement.

SAKIR SEZER received the Dipl.-Ing. degree in electrical and electronic engineering from RWTH Aachen University, Germany, and the Ph.D. degree in 1999 from Queen's University Belfast, U.K. He is currently the Head of Network Security Research with Queen's University Belfast. His research is leading major advances in the field of high performance content processing and is currently commercialized by Titan IC Systems. He is also a Co-Founder and the CTO of Titan IC Systems and a member of several executive committees.